

## Baze de date relaționale

Modelul relațional reprezintă cel mai utilizat model de stocare a datelor, în care datele sunt organizate sub formă de tabele între care există diverse legături.

### Procesul mapării

Transformarea modelului conceptual, a ERD-ului, în modelul fizic, adică în baza de date propriu zisă, se numește mapare. Acest proces implică transformarea fiecărui element al ERD-ului.

Prima etapă a acestui proces constă în crearea tabelelor bazei de date:

- Fiecărei entități îi va corespunde câte un tabel. Spre deosebire de entitate, un tabel va avea numele un substantiv la plural. De exemplu, entitatea **ANGAJAT** se va transforma în tabela **ANGAJATI**.
- Fiecare atribut al unei entități va deveni o coloană a tabelului. Fiecare coloană va memora date de același tip.
- Fiecare instanță a unei entități se va transforma într-un rând(sau înregistrare) a tabelului corespunzător.
- Identificatorul unic al entității devine cheie primară a tabelului. Coloana sau combinația de coloane care identifică în mod unic toate liniile unui tabel se numește **cheie primară**.

Entitati → tabele, (CARTE-carti.dbf)

atribute → campuri, coloane

UID → cheie primara

relatie → cheie straina

business rules → constrangeri

Informațiile despre o tabelă a bazei de date vor fi prezentate folosind **diagramele de tabelă**:

Numele coloanei	Tip	Tip cheie	Opționalitatea
titlu	Varchar2	Pk	*
autor	Varchar2	Pk	*
data_apariției	Date		*
Format	Varchar2		*
Nr_pagini	Number		*

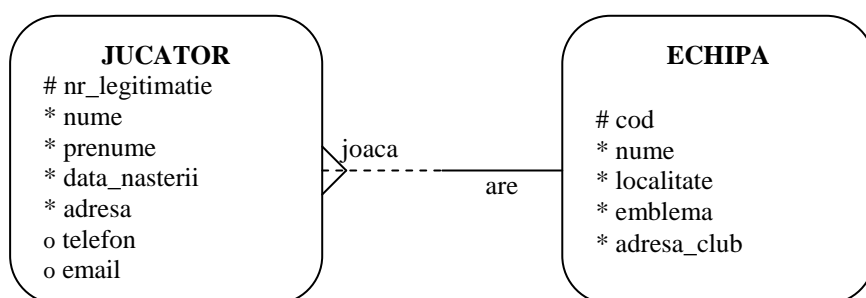
Principalele tipuri de date în Oracle sunt:

Tipul de date	Descriere	Dimensiune Maximă
<b>VARCHAR2</b>	Șir de caractere de lungime variabilă	<b>4000</b> bytes
<b>CHAR</b>	Șir de caractere de lungime fixă	<b>2000</b> bytes
<b>NUMBER (p, s)</b>	Număr având p cifre din care s la partea zecimală. (s negativ reprezintă numărul de cifre semnificative din fața punctului zecimal)	p (precizia) între 1 și 38. s (scala) între -84 și 127.
<b>DATE</b>	Data calendaristică	De la 1 Ianuarie 4712 BC pana la 31 Decembrie, <b>9999 AD</b> .
<b>TIMESTAMP</b>	Se memorează data calendaristică, ora, minutul, secunda și fracțiunea de secundă	Fracțiunea de secundă este memorată cu o precizie de la 0 la 9.
<b>INTERVAL YEAR TO MONTH</b>	perioadă de timp în ani și luni.	

<b>INTERVAL DAY TO SECOND</b>	memorează un interval de timp în zile, ore, minute și secunde	
<b>CLOB</b>	Character Large Object	4 Gigabytes
<b>BLOB</b>	Binary Large Object	4 Gigabytes
<b>BFILE</b>	Se memorează adresa unui fișier binar de pe disc	4 Gigabytes

## Maparea relațiilor one-to-many

Fie ERD-ul din figura următoare:



Observăm că nu putem memora toți jucătorii care joacă la o echipă în cadrul tabelii ECHIPA, deoarece ar trebui să introducem o coloană cu valori multiple. Invers însă, putem cu ușurință să memorăm, pentru fiecare jucător, echipa la care joacă, deoarece acesta nu poate juca decât la o singură echipă.

Astfel, vom memora pentru fiecare jucător codul echipei la care joacă. Deci diagrama de tabel corespunzătoare entității **JUCĂTOR** va fi următoarea:

Numele coloanei	Tip	Tip cheie	Opționalitatea
Nr_legitimatie	Number	Pk	*
Nume	Varchar2		*
Prenume	Varchar2		*
Data_nasterii	Date		*
Adresa	Varchar2		*
Telefon	Number		o
Email	Varchar2		o
<b>Cod echipa</b>	<b>Number</b>	<b>Fk</b>	<b>o</b>

Dacă relația pe partea many este opțională atunci și coloanele cheii străine vor fi opționale. Ce înseamnă acest lucru? Cum un jucător poate la un moment dat să nu joace la nici o echipă, atunci câmpul cod\_echipa va rămâne necompletat în dreptul lui (va avea valoarea NULL). Dacă însă relația este obligatorie pe partea many atunci coloanele ce fac parte din cheia străină vor fi obligatorii.

În general, la maparea unei relații de tip one-to-many, vom introduce în tabela corespunzătoare entității de pe partea many a relației cheia primară a entității de pe partea one a relației. Câmpurile astfel introduse se vor numi **cheie străină** (foreign key).

Așadar:

- cheia străină a unei tabele este cheia primară din tabela referintă
- cheia străină este întotdeauna introdusă în tabela corespunzătoare entității din partea many a relației.

## Maparea relațiilor one-to-one

Dându-se două entități **A** și **B** legate între ele printr-o relație one-to-one, este evident că putem include cheia primară **A** în cadrulul tabelului **B**, dar putem proceda la fel de bine și invers, incluzând cheia primară a tabelului **B** în cadrul tabelului **A**, deoarece fiecărei instanțe a entității **A** îi corespunde cel mult o instanță a entității **B**, dar și invers, oricărei instanțe a entității **B** îi corespunde cel mult o instanță a entității **A**.

Pentru relația din figura I.3.3 de exemplu putem memora pentru fiecare persoană seria de pașaport, dar și invers, pentru fiecare pașaport putem memora cnp-ul deținătorului.

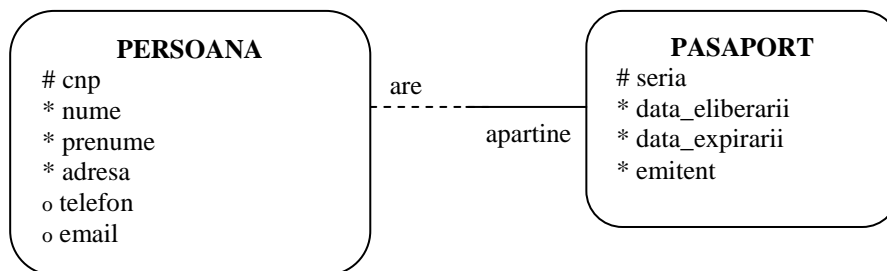


Figura I.3.3.

Decizia depinde de specificul afacerii modelate. Dacă de exemplu ne interesează în primul rând persoanele și abia apoi datele de pe pașapoarte, atunci vom adopta probabil prima variantă, a memorării seriei de pașaport în cadrul tabelului **PERSOANE**, dacă însă baza de date este destinată evidenței pașapoartelor, atunci probabil vom adopta varianta a doua.

Uneori este convenabil să memorăm cheia străină în ambele părți ale relației, în exemplul nostru pentru fiecare pașaport să memorăm cnp-ul persoanei care îl deține, dar și pentru fiecare persoană să memorăm seria de pașaport.

## Maparea relațiilor recursive

Dacă vom privi o relație recursivă ca pe o relație de tipul one-to-many între o entitate și ea însăși, atunci acest caz se reduce la ceea ce deja am discutat. Să exemplificăm relația din figura I.3.4. Relația recursivă din această figură poate fi privită ca o relație între două entități identice, ca în figura I.3.5.

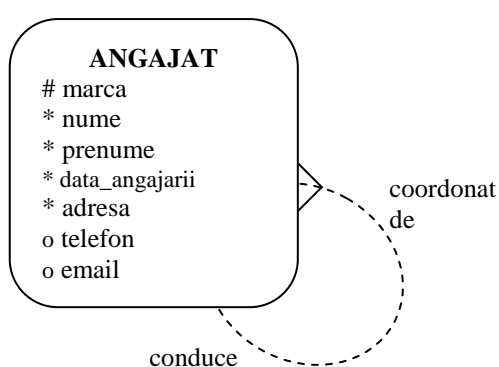


Figura I.3.4.

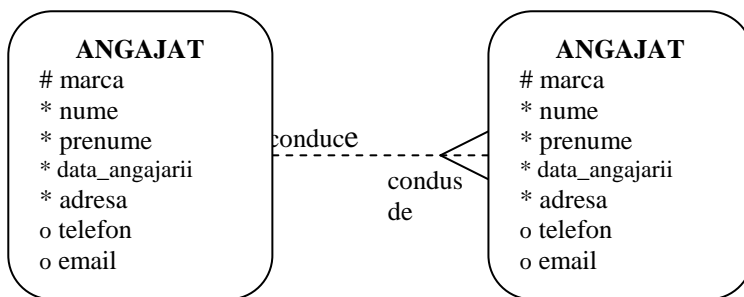


Figura I.3.5.

Așadar vom introduce în cadrul tabelului **ANGAJAȚI**, marca șefului său. Diagrama de tabela va arăta ca mai jos.

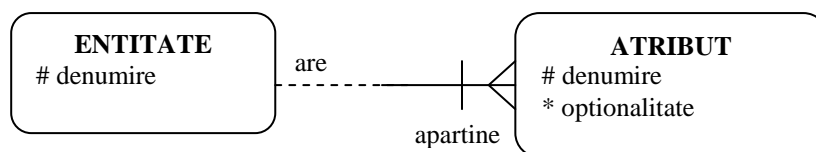
**Tabelul I.3.4.**

Numele coloanei	Tip	Tip cheie	Opționalitatea
Marca	Number	Pk	*
Nume	Varchar2		*
Prenume	Varchar2		*
Data_angajarii	Date		*
Adresa	Varchar2		*
Telefon	Varchar2		o
Email	Varchar2		o
Marca_sef	Number	Fk	o

## Maparea relațiilor barate

Relațiile barate sunt mapate ca cheie străină în tabela aflată în partea many a relației, la fel ca la maparea oricărei relații one-to-many. Bara de pe relație exprimă faptul că acele coloane ce fac parte din cheia străină vor devenii parte a cheii primare a tabelului din partea many a relației barate.

Pentru exemplul din figura I.3.6, cheia primară a tabelului **ATRIBUTE** va fi format din coloanele **denumire\_atribut** și **denumire\_entitate**, aceasta din urmă fiind de fapt cheie străină în tabela **ATRIBUTE**.



**Figura I.3.6.** Maparea relațiilor barate

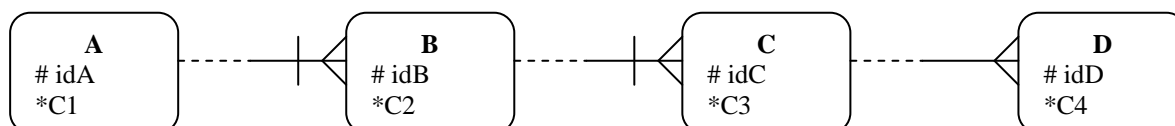
**Tabelul I.3.5.** Tabela **ENTITĂȚI**

Numele coloanei	Tip	Tip cheie	Opționalitatea
denumire	Varchar2	Pk	*

**Tabelul I.3.5.** Tabela **ATRIBUTE**

Numele coloanei	Tip	Tip cheie	Opționalitatea
denumire_atribut	Varchar2	Pk	*
denumire_entitate	Varchar2	Pk, Fk	*
optionalitate	Varchar2		*

Să considerăm acum un exemplu în care există mai multe relații barate, în cascadă.



**Figura I.3.7.** Relații barate în cascadă

**Tabelul I.3.6.** Tabela A

Numele coloanei	Tip cheie	Opționalitatea
<b>idA</b>	<b>Pk</b>	*
C1		*

**Tabelul I.3.7.** Tabela B

Numele coloanei	Tip cheie	Opționalitatea
<b>idB</b>	<b>Pk</b>	*
C2		*
<b>idA</b>	<b>Pk, Fk</b>	*

**Tabelul I.3.8.** Tabela C

Numele coloanei	Tip cheie	Opționalitatea
<b>idC</b>	<b>Pk</b>	*
C3		*
<b>idA</b>	<b>Pk, Fk</b>	*
<b>idB</b>	<b>Pk, Fk</b>	*

**Tabelul I.3.9.** Tabela D

Numele coloanei	Tip cheie	Opționalitatea
<b>idD</b>	<b>Pk</b>	*
C4		*
<b>idA</b>	<b>Fk</b>	*